

Amendments to the Claims

This listing of claims will replace all prior versions and listings of claims in the application:

Listing of Claims

1. (previously presented) A display system, comprising:

a display;

a display buffer coupled to the display; and

a processor adapted to execute an application program which, when executed, produces images upon the display, wherein during a first mode the images are forwarded in sequence to the display, and wherein during a second mode the images are compiled as a combination image of at least one of said images drawn over at least another of said images and presented to the buffer before being forwarded to the display.

2. (previously presented) The display system as recited in claim 1, wherein the application program disables or enables buffering of the images by configuring the processor to execute in either the first or second mode.

3. (original) The display system as recited in claim 1, wherein the images comprise frame, panel and button images.

4. (original) The display system as recited in claim 1, wherein the application program comprises a Java program.

5. (previously presented) A computer-readable storage medium, comprising:

an operating system,

an application program running on code compatible with the operating system; and

a software component invoked by the application program to display object code which, when executed, produces a sequence of images upon a display screen, wherein the software component can be configured during runtime of the application program to enable or disable buffering of the sequence of images as a combination image before sending the combination image to the display.

6. (currently amended) The ~~memory~~ medium as recited in claim 5, wherein the object code is part of a graphical user interface associated with the application program.

7. (currently amended) The ~~memory~~ medium as recited in claim 5, wherein the software component comprises an application program interface of code which translates between code within the application program and the operating system.

8. (currently amended) The ~~memory~~ medium as recited in claim 7, wherein a behavior of the application program interface emulates that of a second application program interface based on windowing protocols of a windows-based version of said operating system.

9. (currently amended) The ~~memory~~ medium as recited in claim 8, wherein the second application program interface comprises a Java abstract windowing toolkit.

10. (currently amended) The ~~memory~~ medium as recited in claim 5, wherein the application program is written in Java programming language.

11. (currently amended) The ~~memory~~ medium as recited in claim 5, wherein the operating system comprises a Windows, Unix or OS/2 computer operating system.

12. (previously presented) A method for displaying images of objects upon a display, the method comprising:

providing an application program running under an operating system;

creating the images of the objects using an interface independent of the operating system;
and

enabling or disabling buffering of said images during runtime as directed by the application program, wherein:

during a first mode, said buffering is disabled so that the images are forwarded in sequence to the display; and

during a second mode, said buffering is enabled so that the images are compiled as a combination image of at least one of said images drawn over at least another of said images and presented to the buffer before being forwarded to the display.

13. (original) The method as recited in claim 12, wherein said creating comprises compiling the object as code that includes part of a graphical user interface associated with the application program.

14. (original) The method as recited in claim 12, wherein said creating comprises implementing a call routine to compile a software component that includes an application program interface between the application program and the operating system.

15. (previously presented) The method as recited in claim 14, wherein a behavior of the application program interface emulates that of a second application program interface based on the operating system.

16. (previously presented) The method as recited in claim 15, wherein the second application program interface comprises a Java abstract windowing toolkit.

17. (original) The method as recited in claim 12, wherein the application program is written in Java programming language.

18. (original) The method as recited in claim 12, wherein the operating system comprises a Windows, Unix or OS/2 computer operating system.

19. (previously presented) A computer-readable storage device, comprising:

a windows-based operating system;

an application program running under the operating system;

a plurality of objects created at runtime by the application program;

an interface independent of the operating system, and adapted for:

creating images of the objects; and

enabling or disabling buffering of the images to a memory storage area prior to displaying the images, as directed by the application program, wherein:

during a first mode, said buffering is disabled so that the images are forwarded in sequence to a display; and

during a second mode, said buffering is enabled so that the images are compiled as a combination image of at least one of said images drawn over at least another of said images and presented to the memory storage area before being forwarded to the display.

20. (previously presented) The display system as recited in claim 2, wherein the processor executes in the first mode when the display is directly coupled to the processor.

21. (previously presented) The display system as recited in claim 2, wherein the processor executes in the second mode when the display is remotely coupled to the processor.

22. (previously presented) The computer-readable storage device as recited in claim 19, further comprising a software component associated with the interface and adapted to: (i) determine whether the application program is operating in a remote or direct mode, and (ii) create a peer component to enable or disable buffering of the graphical representation of the object based on the determination made by the software component.